0 Einführung ins Thema

0.1 Logik als "Fundament der Mathematik"

Hilberts Programm (ca. 1900–1928, initiiert von David Hilbert)

Ziel: formale Grundlegung der Mathematik

Mittel: mathematische Logik:

- mathematische Strukturen als logische Strukturen
- mathematische Aussagen als logische Formeln
- mathematische Beweise durch "syntaktisches Schließen" (Symbolmanipulation: Axiome, Schlussregeln)

Ansatz: Rückführung der Mathematik auf **Arithmetik und Mengenlehre**

Zwei Kernfragen:

(1) Kann jede mathematische Aussage durch mathematisches Schließen bewiesen oder widerlegt werden?

(2) Gibt es ein Verfahren, das zu jeder mathematischen Aussage entscheidet, ob sie wahr oder falsch ist?

Beachte: Es gilt (1) $\Longrightarrow$ (2)

Eine äquivalente Formulierung des **Entscheidungsproblems** (2) ist das **Allgemeingültigkeitsproblem** der Logik erster Stufe:

Allgemeingültigkeitsproblem der Logik erster Stufe	
Eingabe:	Eine Formel φ der Logik erster Stufe
Frage:	Gilt für alle zu φ passenden Interpretationen $\mathcal{I}$: $\mathcal{I}$ erfüllt φ?

Beispiel: Sei φ die Formel

$$\begin{aligned}\forall x\,\exists y\,\exists z\,\Big(&x \leq y \;\wedge\; z = y+1+1 \;\wedge\\ &\forall u\,\forall v\,\Big(\big(u \times v = y \;\rightarrow\; (u=1 \;\vee\; v=1)\big) \;\wedge\\ &\qquad\big(u \times v = z \;\rightarrow\; (u=1 \;\vee\; v=1)\big)\Big)\Big).\end{aligned}$$

Beachte: φ besagt "es gibt unendlich viele Primzahlzwillinge".

Der Nachweis, dass die Formel φ in der Arithmetik der natürliche Zahlen erfüllt ist, würde also ein berühmtes offenes Problem aus der Zahlentheorie lösen — nämlich die Frage, ob es unendlich viele Primzahlzwillinge gibt.

Zwei wichtige Aspekte der logischen Fundierung:

(1) Präzisierung von Aussagen (*"Logik für Penible"*)

(2) Automatisierung des Beweisens (*"Logik für Faule"*)

Zwei Spielverderber:

(1) **Kurt Gödel (1931)**
- $+$: jede gültige Aussage kann durch syntaktisches Schließen bewiesen werden (**Vollständigkeitssatz**)
- $-$: in der Arithmetik gibt es Aussagen, die weder beweisbar noch widerlegbar sind (**Unvollständigkeitssatz**)
- $\Rightarrow$ Hilberts (1) funktioniert nicht!

(2) **Alan Turing (1936)**
- $+$: Der Begriff "automatisch entscheiden" lässt sich einfach und sauber definieren ($\rightsquigarrow$ **Turingmaschine**)
- $-$: Für die Arithmetik gibt es kein automatisches Verfahren — sie ist unentscheidbar
- $\Rightarrow$ Hilberts (2) funktioniert nicht!

Logik und Mathematik: Geschichte

um 325 v. Chr.:	• Aristoteles: Syllogismen • Euklid: Versuch einer Axiomatisierung der Geometrie
um 1700:	Leibniz formuliert das Ziel einer universellen Sprache zur Formulierung aller mathematischen Aussagen und eines Kalküls zur Herleitung aller wahren Aussagen.
um 1850:	Axiomatisierung der Analysis
1854:	Boole: Formalisierung der Aussagenlogik
1879:	Frege: Formalisierung der Logik erster Stufe
um 1880:	Cantorsche Mengenlehre, Rückführung der Analysis und Arithmetik auf die Mengenlehre
im 1900:	Antinomien: Cantorsche Mengenlehre führt zu Widersprüchen ($\nearrow$ *"die Menge aller Mengen, die sich nicht selbst als Element enthält"*) $\Longrightarrow$ Notwendigkeit einer neuen Grundlegung der Mathematik/Mengenlehre

um 1900:	Hilberts Programm. Ziel: • Formalisierung der Mathematik • Nachweis der Widerspruchsfreiheit der Arithmetik
um 1910:	Russel, Whitehead: Mengenlehre mit Typen
um 1920:	Zermelo, Fraenkel: Axiomatische Mengenlehre
1930:	Gödels Vollständigkeitsatz
1931:	**Gödels Unvollständigkeitsätze**
1936:	**Church/Turing**: Es gibt kein Programm, das für alle mathematischen Aussagen entscheidet, ob sie wahr oder falsch sind.

0.2 Logik in der Informatik

Anwendungsbereiche der Logik in der Informatik

- Logische Programmierung
- automatisches Beweisen
- Programm-Verifikation
- **Model Checking** (automatische Verifikation)
- **Logik als Datenbank-Anfragesprache**

0.2.1 Model Checking

Zwei Beispiele zur Motivation:

(1) Der Pentium-Fehler

Pentium-Prozessor (1993):

- Zur Effizienz-Steigerung der Division wurden Wertetabellen verwendet.
- ABER: 5 Einträge waren falsch!
  $\rightsquigarrow$ ca. 1 Fehler je 9 Milliarden Divisionen
  ($\Rightarrow$ Fehler durch "Testen" nicht leicht zu finden)

Kosten: ca. 475 Millionen Dollar
Intel hat danach viele Experten für automatische Verifikation gesucht!

(2) Die Ariane 5-Rakete

Messwerte wurden von 64-Bit-Zahlen in 16-Bit-Zahlen umgewandelt.

- Das hatte bei Ariane 4 gut funktioniert
- ABER: aufgrund der technischen Änderungen waren die Werte bei Ariane 5 größer als erwartet
  $\rightsquigarrow$ Überlauf! System schaltete sich ab, Backup-System übernahm
  Problem: dort lief aber das selbe Programm

Kosten: ca. 600 Millionen Euro

Prinzip der automatischen Verifikation

(1) Modelliere das zu testende System durch ein **Transitionssystem** $\mathcal{T}$ (eine bestimmte logische Struktur; ein beschrifteter Graph).

(2) Drücke die (erwünschte oder unerwünschte) Systemeigenschaft durch eine Formel φ einer geeigneten Logik aus.

(3) Teste, **ob $\mathcal{T}$ die Formel φ erfüllt**.

0.2.2 Logik als Grundlage für Datenbank-Anfragesprachen

Grundprinzip:

- Datenbank $\widehat{=}$ logische Struktur $\mathfrak{A}$
- Anfrage $\widehat{=}$ Formel φ einer geeigneten Logik
- Auswerten der Anfrage auf der Datenbank $\widehat{=}$ Testen, ob "$\mathfrak{A}$ erfüllt φ" gilt

Details: Vorlesung "Diskrete Modellierung"

0.3 Literaturhinweise

Dieses Kapitel orientiert sich an Kapitel 1 aus [Sch07]. Zur weiteren Lektüre seien jeweils das erste Kapitel und die Einleitung von [EFT07] und [Lib04] empfohlen. Der Artikel [HHI+07] bietet einen Überblick über verschiedene Anwendungsgebiete der Logik in der Informatik.